

Guide to installing and running ROS

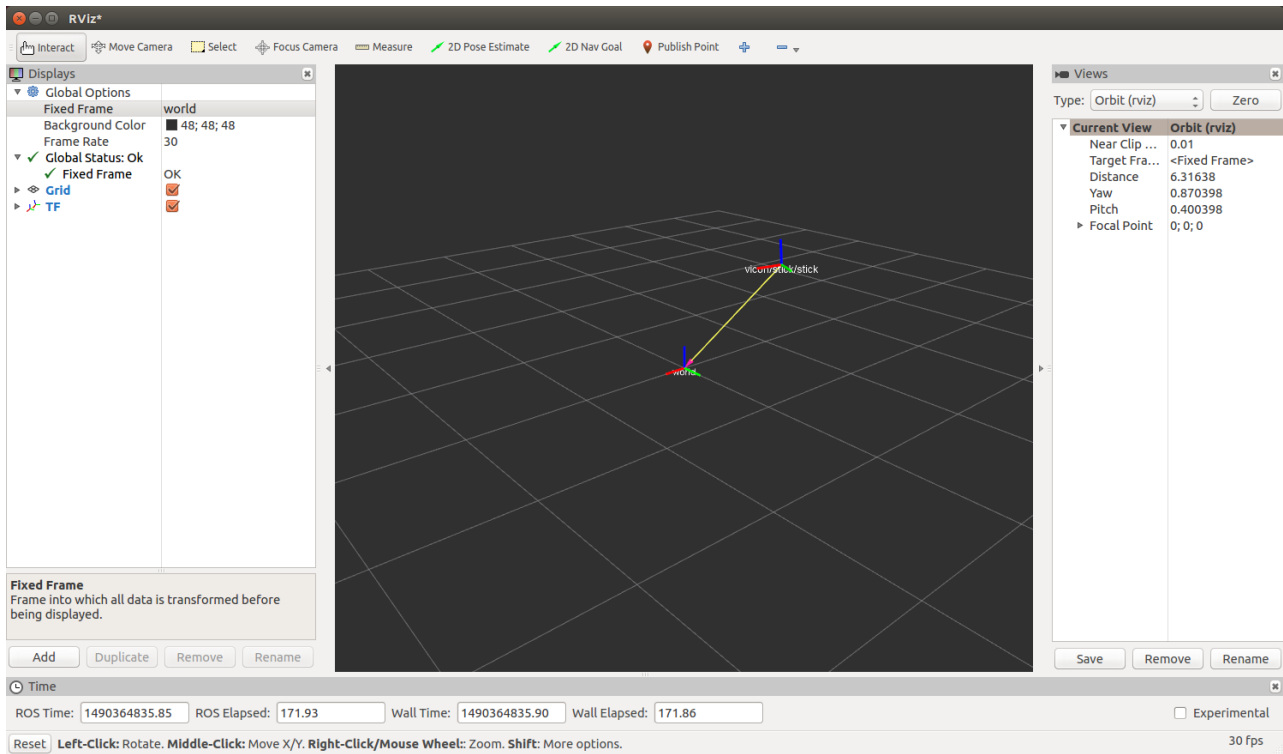
Support

Exported on 08/01/2023

Table of Contents

1 Vicon	4
2 Linux	5
3 ROS	6
4 Setup catkin workspace for ROS.	7
5 Vicon_bridge	8
6 RVIZ	9
7 Re-Running after computer shutdown	11
8 Related articles	14
9 Additional Notes - 2022 Update	15

This is a quick guide to setting up and getting ROS to run on a Linux. The guide shows you how to get the data from a networked machine and run it through the `vicon_bridge` created by ETH Zurich and into RVIZ. All the bridge does is take the information and convert it to the correct format for ROS. Then RVIZ takes that data and shows a 3d representation of the objects moving around. This is just the basic functions of the ROS setup. This guide does not cover path finding setup or anything outside of the Vicon data.



1 **Vicon**

Make note of the IP address for your Vicon server

2 Linux

Make note of boost lib using:



code

```
dpkg -s libboost-dev | grep 'Version'
```

Make note of Linux version". I used 14.04

3 ROS

Before you install anything else you need to install ROS on the Linux computer. First thing, make your way to the ROS install page (<http://wiki.ros.org/kinetic/Installation/Ubuntu>) or make your way to this guide (<http://people.oregonstate.edu/~chungje/Code/Pioneer3dx%20simulation/ros-indigo-gazebo2-pioneer.pdf>). follow only the first section of you follow the guide from Oregon.

ROS stands for Robot Operating System. it is used by most every robot so there is a common language between them. Most places working with robots will be using this OS

First thing, you have to get your computer to accept the ROS package. Open up a terminal and run:

i code

```
sudo sh -c 'echo "deb http://packages.ros.org/ros/ubuntu $(lsb_release -sc) main" > /etc/apt/sources.list.d/ros-latest.list'
```

Get a validation key for the ROS install:

i code

```
wget https://raw.githubusercontent.com/ros/rosdistro/master/ros.key -O - | sudo apt-key add -
```

Update and install the package:

i code

```
sudo apt-get update
sudo apt-get install ros-indigo-desktop-full
```

Initialize the rosdep

i code

```
sudo rosdep init
rosdep update
```

Setup the environment for ROS:

i code

```
echo "source /opt/ros/indigo/setup.bash" >> ~/.bashrc
source ~/.bashrc
```

Finally get rosininstall:

i code

```
sudo apt-get install python-roscpp
```

4 Setup catkin workspace for ROS.

This is where you will work on any ROS projects. This is also where you would put any packages downloaded and run them (e.g. vicon_bridge).

Open up a new terminal and make the directory and navigate to it:

```
i code  
mkdir -p ~/catkin_ws/src  
cd ~/catkin_ws/src
```

Initialize the workspace and navigate again:

```
i code  
catkin_init_workspace  
cd ~/catkin_ws
```

Then build your workspace and add to your path:

```
i code  
catkin_make  
echo "source /home/user_name/catkin_ws/devel/setup.bash" >> ~/.bashrc  
source ~/.bashrc
```

Source the new setup:

```
i code  
source devel/setup.bash
```

Now you can add your packages to the workspace

5 Vicon_bridge

This is code written by ETH Zurich to push the Vicon data through to the odometry coordinates. It will be needed for any system in ROS for the data to be read. This is a great point to start if you are just starting with ROS. It used SDK version 1.3

Found here: http://wiki.ros.org/vicon_bridge

Code found here: https://github.com/ethz-asl/vicon_bridge

Download the master folder from the github location and extract the contents.

Then move the file to the src folder in to the workspace:

i code
~/catkin_ws/src

Build and run the code:

i code
cd ~/catkin_ws
catkin_make
source devel/setup.sh

Launch the program:

i code
roslaunch vicon_bridge vicon.launch

At this point, you should see it start running. But you will notice it will try connecting to Vicon:801. you will need to stop the code and edit a file to make it work.

i stopping the code
Stop the program running with ctrl+C

To fix this, make your way to the file vicon.launch in ~/catkin_ws/src/vicon_bridge-master/launch and open the file vicon.launch and change the value related to datastream_hostport to the value of the IP of the host computer (e.g. 192.168.0.254) keep the port number to 801 as that is the standard port for vicon to use.

At this point, you can get the vicon data into the ROS side of things. If you want to visualize the data continue to the next step

6 RVIZ

RVIZ is a tool to visualize the data in a 3d environment. it can also be used to connect models, set waypoints and display information.

Found here: <http://wiki.ros.org/rviz/UserGuide>

To install rviz (I installed indigo version as that is the version of ROS I am using) open up a new terminal and run:

i **code**
`sudo apt-get install ros-indigo-rviz`

Build from source and install:

i **code**
`rosdep install rviz`

Build the visualizer:

i **code**
`rosmake rviz`

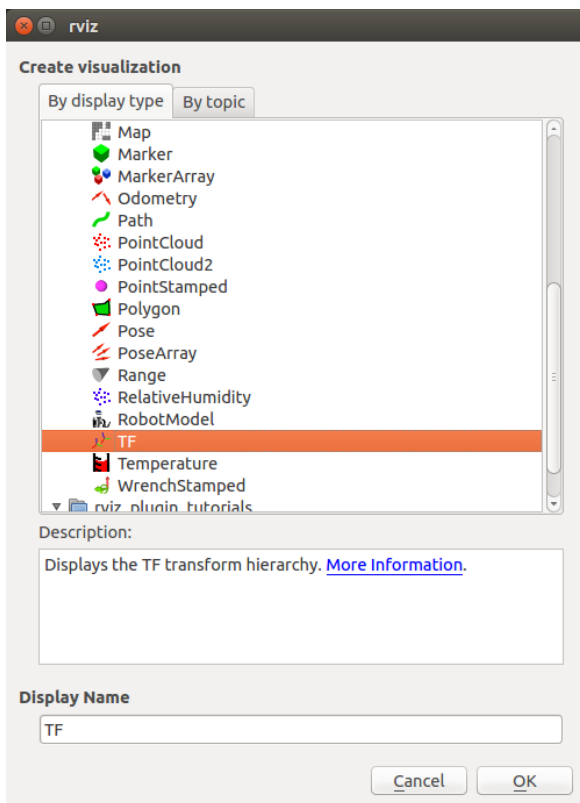
Then start rviz:

i **code**
`source /opt/ros/indigo/setup.bash`
`roscore &`

Start the visualizer:

i **code**
`roslaunch rviz rviz`

You should now see rviz running in a window. If you run the updated vicon_bridge, you should see that the section that says "global status" should have a warning. If you change the "Fixed Frame" section to show "world", you will see the global status change to "OK". After that you just need to add the Transforms into the window. Do that by pressing the "Add" button and then select "TF". You should now see the objects from the datastream in the 3d view.



If you are not getting data to the software, check firewalls on both computers and make sure that the data can get through.

You will want to keep the terminals for the rviz and vicon_bridge open as ctrl+c in those windows stops the processes and closes them. If you just close the program, it may not end the processes.

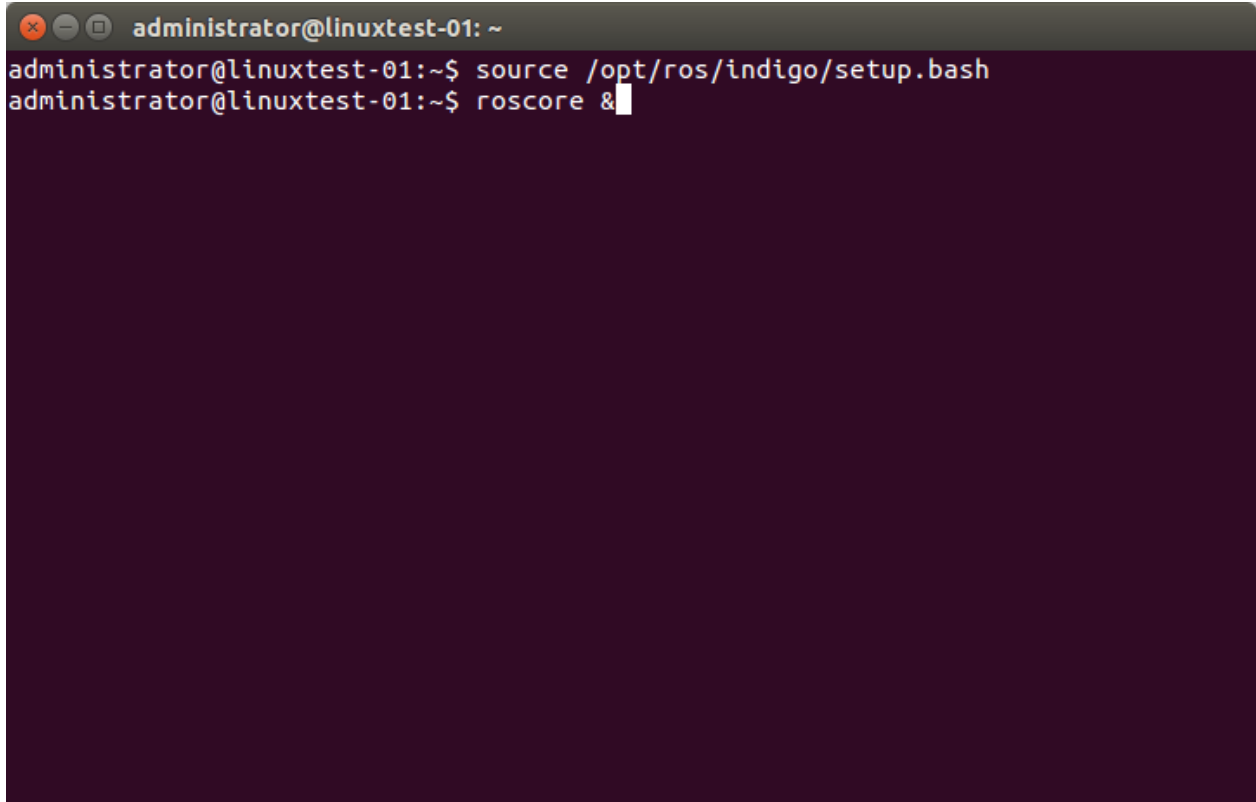
7 Re-Running after computer shutdown

after you restart your computer, there are a couple things you have to do to re-run each of the parts.

re-initialize the source command for Rviz and start roscore

code

```
source /opt/ros/indigo/setup.bash  
roscore &
```



A terminal window titled 'administrator@linuxtest-01: ~' with a dark purple background. It shows the following commands and their prompts:

```
administrator@linuxtest-01:~$ source /opt/ros/indigo/setup.bash  
administrator@linuxtest-01:~$ roscore &
```

run rviz:

code

```
roslaunch rviz rviz
```

```
roscore http://linuxtest-01:11311/
Press Ctrl-C to interrupt
Done checking log file disk usage. Usage is <1GB.

started roslaunch server http://linuxtest-01:52969/
ros_comm version 1.11.20

SUMMARY
=====

PARAMETERS
* /rostdistro: indigo
* /rosversion: 1.11.20

NODES

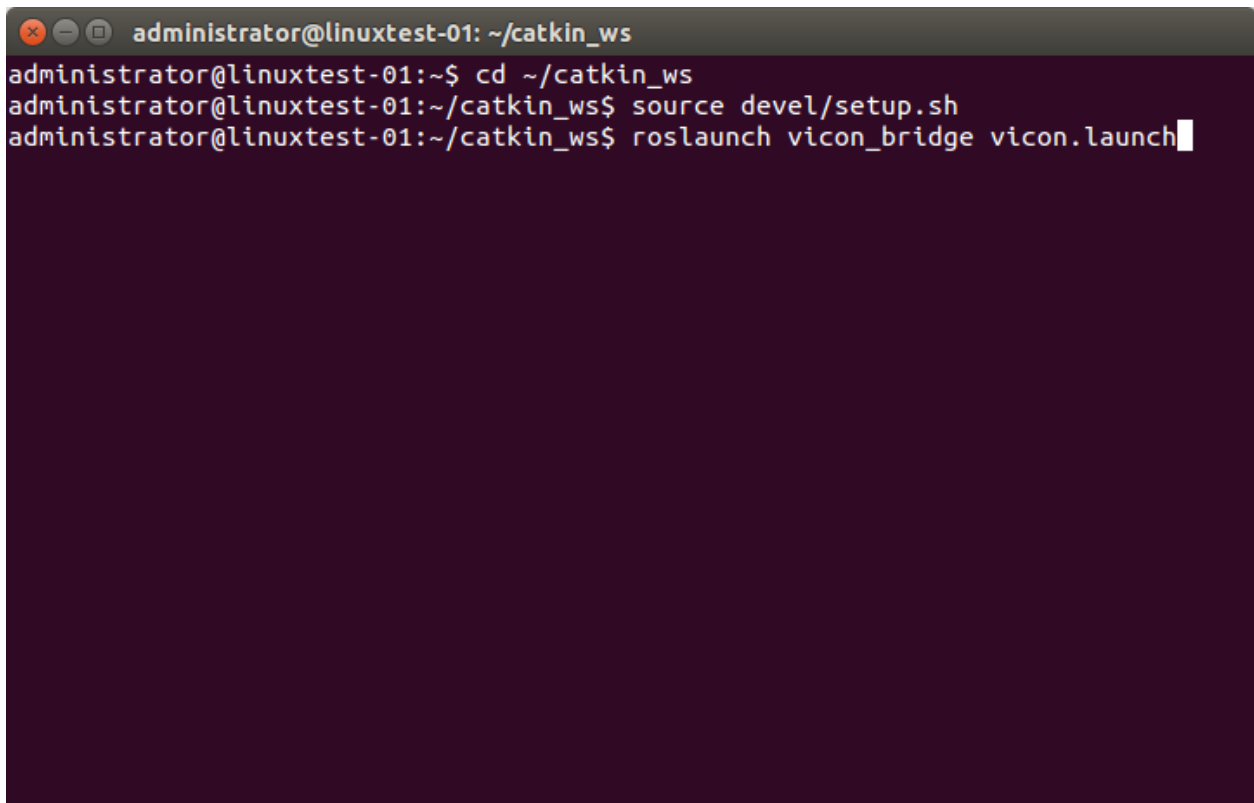
auto-starting new master
process[master]: started with pid [10070]
ROS_MASTER_URI=http://linuxtest-01:11311/

setting /run_id to c948e6d2-10a4-11e7-9c4a-001517c536a4
process[rosout-1]: started with pid [10083]
started core service [/rosout]
roslaunch rviz rviz
```

Now you need to navigate, re-initialize and run the vicon_bridge

code

```
cd ~/catkin_ws
source devel/setup.sh
roslaunch vicon_bridge vicon.launch
```

A terminal window with a dark background and light-colored text. The window title bar shows three icons (close, minimize, maximize) and the text 'administrator@linuxtest-01: ~/catkin_ws'. The terminal content shows three lines of commands and their prompts: 'administrator@linuxtest-01:~\$ cd ~/catkin_ws', 'administrator@linuxtest-01:~/catkin_ws\$ source devel/setup.sh', and 'administrator@linuxtest-01:~/catkin_ws\$ roslaunch vicon_bridge vicon.launch' followed by a cursor.

```
administrator@linuxtest-01: ~/catkin_ws
administrator@linuxtest-01:~$ cd ~/catkin_ws
administrator@linuxtest-01:~/catkin_ws$ source devel/setup.sh
administrator@linuxtest-01:~/catkin_ws$ roslaunch vicon_bridge vicon.launch
```

8 Related articles

 [Guide to installing and running ROS](#)(see page 3)

9 Additional Notes - 2022 Update

This page is an addendum on Owen's instructions to reflect the state of the Denver Linux PC in 2022.

The following will try to outline all changes:

- Operating System: 20.04.07 (original: 14.04)
- ROS Distribution: noetic (original: indigo)
 - anywhere where it says indigo in the instructions, replace with kinetic
 - the ROS distribution depends on the OS so update as necessary

Initiating from PC Shutdown:

- First check the vicon.launch file that it is referencing the Tracker PC:
 - located here: /home/owen/catkin_ws/src/launch
- When initiating ROS from shutdown, open a Terminal and use the following two commands:

```
source opt/ros/kinetic/setup.bash
roscore &
```

- If you want to visualize in rviz, still run the same command

```
roslaunch rviz rviz
```

- Open a new terminal. In the original instructions, it required you to navigate, re-initialize and run the vicon_bridge. However, there is an issue and you will need to run the following command
 - this may be specific to the Denver PC
 - this updates any outdated packages and dependencies
 - when prompted, enter the password: Vicon123 (does not display)

```
sudo apt-get upgrade
```

- Now you can navigate, re-initialize and run the vicon_bridge. Namely:

```
cd ~/catkin_ws
source devel/setup.sh
roslaunch vicon_bridge vicon.launch
```

- If you had rviz open, you should now be able to choose `vicon_world` in the fixedframe field
- If your object disappears, click reset in the bottom left

To run CrazySwarm

Navigate to project (crazyswarm)

```
cd crazyswarm
```

Configuring joystick

<http://wiki.ros.org/joy/Tutorials/ConfiguringALinuxJoystick>

```
cd /home/owen/crazyswarm/ros_ws/src/crazyswarm/launch
```

<https://crazyswarm.readthedocs.io/en/latest/configuration.html#adjust-configuration-files>

<https://www.bitcraze.io/documentation/tutorials/getting-started-with-crazyflie-2-x/#flying>